Critical Design Review Package

Team Name: Date: December 7, 2005

Project Leader: Steve Wu

Name Signature

Matthew Martinez ______________________________

Pablo Rosado ______________________________

Steve Wu ______________________________

Table of Contents

Project Organization Chart

Written by Steve Wu

Project Team Structure

Written by Steve Wu

Project Overview

Written by Pablo Rosado

The purpose of modern information technology is to make things more convenient, it has been the basis of the technology market, which simply put is to create more efficient medium of accessing information. Using the same principle and utilizing technology that has been developed, autonomous or self driving devices can be used to further increase the convenience that technology provides. The concept of the BlueKey project is to achieve just that, the team intends to implement a reactive system that will cause other systems to respond in a customizable way. Now most people tend to perform the same tasks day in and day out, it is conceivable that some of those tasks can be automated through technology. For example if a person's daily routine is to come home get a snack, sit down at there computer, log in, check there email, and browse a few websites. It would be far more convenient for that person for these tasks to occur when they enter their home, that way by the time they sit down in front of there computer everything is ready for them. Now the main difference between BlueKey and the few other systems out there like it is the use of the Bluetooth wireless protocol; which provides an advantage since the protocol is designed to support multiple devices that are sold on the consumer market. The device that BlueKey is developing is a small handheld device that can not only store information but also transmit commands to a specific device when it's in range. The devices ultimate goal is to be similar to a USB thumb drive that has Bluetooth connectivity as an additional feature. Where it can then be sold to numerous markets depending on what software is used in conjunction with it. Hence the software aspect makes the devices potential limitless.

Specifications and Interface Requirements

Bluetooth Protocol
Written by Steve Wu

Using Bluetooth devices that follow the following requirements:

SIG certified device
Bluetooth Version 1.1 or 1.2
Class 2 device
Range 10 meters
Frequency 2.4 GHz
Bandwidth 100Mbps

Bluetooth Module and Module Commands
Written by Matthew Martinez

The Bluetooth module will interface with the flash memory through parallel communication over its multiple programmable I/O pins.

The Bluetooth module will be programmed to respond to the following string commands from the computer:

String Command
Action
ID_REQ
Send ID
AK_REQ

Reply with Acknowledgment
NA_REQ
Reply
BY_REQ
Disconnect
S0_REQ ... S9_REQ
Access Internal Memory
S1_EXREQ … S32_EXREQ*
Access External Memory

*These commands will be replaced with commands that perform more specific actions.

Memory
Written by Pablo Rosado

The memory we chose for the project is the Atmel 64 megabit dual interface data flash chip (AT45DB642). The memory of the AT45DB642 is divided into three levels consisting of sectors, blocks, and pages. Each sector contains a block or multiple blocks with a size or 8448 bytes each and within each block there consist of 8 pages with a size of 1056 bytes each. In total, there are 8192 pages in the memory. The main memory is divided into 33 sectors (Sector 0 – Sector 32). Sector 0 contains only one block and that block contains only 8 pages. Sector 1 contains 31 blocks and 248 pages. The remaining sectors (Sector 2 – Sector 32) contain 32 blocks each and 256 pages. In addition to the main memory, the AT45DB642 contains two SRAM data buffers of 1056 bytes each which is also equivalent the size of a page.

Figure 1: Memory Architecture

To achieve the first purpose of the external memory, to store a Bluetooth Mac address with personal data that corresponds to the automation features, we will make use of Sector 0 of the memory. In Sector 0 in the memory array, the Bluetooth Mac Address which is 48 bits long will be stored in block 0 in page 0. However, because this information will not take up the entire page, we will add irrelevant Hex values to fill in the remaining space. The remaining pages in Sector 0 will be use to create an index for every page in the memory array from Sector 1 to Sector 32. Sector 1 will be used to store the data relevant to the automation features. The remaining sectors of the memory, Sector2 through Sector 32, will be use for storing personal data of any type to fulfill the second purpose of the external memory, allowing BlueKey to act as a personal thumb drive.

The AT45DB642 will be directly connected to the Bluetooth module (WML-C10) through the dedicated parallel interface (I/O7-I/O0: pins 26 – 29 and pins 32-35) in order to increase speed performance rather than using the dedicated serial interface which can only process one bit at a time. This will occur when the SER/PAR pin (pin 25) of the data flash chip is held low. From there, the Bluetooth module will communicate to the computer wirelessly.

When a connection has been establish between the computer and the Bluetooth module, the address index for the pages of the main memory will be created and sent to the module after completion. The module afterwards will send the address index to the computer which it will store on its memory. The purpose of this is to allow the computer to keep track and know the exact location where the data it has written to the external memory.

Writing data to the main memory directly is not allowed and can only be done through either using buffer 1 or buffer 2. As result, all data that is sent from the computer to the Bluetooth module will be first written into the buffer first and then will be written into the main memory. Note for this process we will be only making use of buffer 1. When the computer specifies to the module a request to write data from the main memory, the chip select pin (pin 15) is low and an opcode of 83H (this specifies the use of buffer 1 only) will be clocked into the device followed by three address bytes consisting of 13 page address bits that specify the page in the main memory to be written and 11 don't care bits. During this process, the chip select pin will go through a low to high transition which will cause an erase of data of the selected page in the main memory and then write the data stored in the buffer into that specified page. Writing data into the buffer requires a 1 byte opcode of 83H to be clocked into the device followed by three address bytes consisting of 13 don't care bytes and 11 buffer address bits. The 11 buffer address bit specifies the first byte in the buffer to be written. After the last address byte has been clocked into the device, data can then be clocked in on subsequent clock cycles. When writing data to the buffer is completed, the chip select pint will go through a low to high transition to indicate there is no more data to be written.

When the computer specifies to the module a request to read data from the main memory, the chip select pin (pin 15) is low and an opcode of 52H will be clocked into the device followed by three address bytes which includes the 24 bit page, the byte address sequence and a series of 60 don't care bytes. The first 13 bits of the 24 bit address sequence specifies the page in the main memory to be read. The last 11 bits specify the starting byte address within that page. The 60 don't care bytes will be use to initialize the read operation. Afterwards, additional pulses from the clock (SCK/CLK, pin 16) will send the data from the specify page to be outputted to the parallel output pins (I/O1-I/O7) which then will be sent to the Bluetooth module (WML-C10) to send to the computer in packets of 1KB in size. When the read operation is done, the chip select pin is set on high.

Erasing data from the main memory can be done on a page basis. When the computer indicates to the module to delete data from the main memory, an opcode of 81H must be loaded into the device followed by three address bytes consisting of 13 page address bits and 11 don't care bits. When a low to high transition occurs on the chip select pin, the specified page will be erased. Blocks of information within a sector can also be erased. To perform a block erase, an opcode of 50H must be loaded into the device followed b three address bytes consisting of 10 page address bits and 14 don't car bits. The 10 page address bits are used to specify which block to erase. When a low to high transition occurs on the chip select pin, the memory will erase the indicated block of memory.

Power System

Written by Matthew Martinez

Bluetooth Module Power Requirements:

Voltage: 1.8 V and 3.3 V
Current: 60 mA

Memory Power Requirements:
Voltage: minimum 2.7 V
Current: 20 mA

The device must possess a switch to turn the device on and off.
The power source will consist of coin batteries.
The device will possess a power indicator.
The regulation circuits will utilize the least number of components possible to minimize the size of the device.

Operating Conditions:
Temperature: about -30°C (-22°F) to about 60°C (140°F)
Battery life: 500 hours (about 20 days)

Software System Requirements
Written by Steve Wu

Operating System: Windows 2000/XP
Processor Platform: x86 processor
File System: FAT32 or NTFS
RAM: 256MB
Optional: USB 1.0 or higher support

Bluetooth Interference Performance Specifications
Written by Pablo Rosado

The strength and value of any wireless system can be measured in various ways, from data transfer speed to distance range. However, interference and integrity degradation is perhaps the most important issue to be considered when evaluating the performance of any wireless system. BlueKey uses the Bluetooth wireless technology in order to communicate to other devices. Natural interference such as a wall, a door, or moving objects such a people or animals is not a real problem with a Bluetooth device. Since Bluetooth uses the 2.4 GHz ISM bandwidth (a popular and free frequency use by many wireless devices), the more immediate threat of interference lies with other wireless devices including multiple Bluetooth devices being in use.

Interference with multiple Bluetooth Piconets:
When a Bluetooth device encounters interference on a channel, it deals with the problem by hopping to the next channel and trying again or retransmitting the last data packet. In this manner it can attempt to avoid interference with another wireless device completely. In the case of multiple Bluetooth networks operating at once while being physically located in close proximity to one another, the probability of frequency collision increases and the time for each Bluetooth network to find a free channel also increases. The figure below demonstrates the probability of frequency collision occurring with multiple Bluetooth piconet occurring at once is a large open room.

This figure above illustrates annoying interference will start to occur once the number of piconets grow to 15 and beyond the user will receive 10% less data than expected.

The figure above shows the probability of frequency occurring in a large office separated by partitions (cubicles). The probability of interference naturally grows as more adjacent piconets become active within the ten-meter range as illustrated in both graphs. When interference occurs, the Bluetooth data channel applies retransmission and can therefore cope with interference, however at the performance threshold 10% probability of error or dropped packet, the user will experience annoying interference. In an interference free channel the DH1 data transfer rate is 172.8 kb/s. The figure below illustrates the data degradation, which results from collisions and retransmits. As 47 collisions occur, although packets are retransmitted, the performance results degrade over time. The graph provides information on the typical interference affecting a Bluetooth device operating in a multi-Bluetooth scenario, and its effects on the capacity.

Interference with Wireless LAN:

Because both Wi-Fi and Bluetooth wireless technology operate in the same frequency spectrum and will often be located in close physical proximity to one another, it is clear that interference of some degree will occur under certain scenarios.

Wi-Fi uses Direct Sequence Spread Spectrum (DSSS) instead of FHSS. Its carrier does not hop or change frequency and remains centered on one channel that is 22 MHz-wide. When a Bluetooth radio and a Wi-Fi radio are operating in the same area, the single 22 MHz-wide Wi-Fi channel occupies the same frequency space as 22 of the 79 Bluetooth channels which are 1 MHz wide. When a Bluetooth transmission occurs on a frequency that lies within the frequency space occupied by a simultaneous Wi-Fi transmission, some level of interference can occur, depending on the strength of each signal.

Wi-Fi acts like a wireless Ethernet, and it deals with interference like Ethernet does. If a transmission fails it assumes that a collision has occurred due to two stations trying to transmit simultaneously, and an ARQ is issued. In addition, many installations of 802.11b utilize the optional automatic data rate modification feature. This allows the data rate to fall back from 11 Mbps to 5.5, 2, or even 1 Mbps, in an effort to lower the Bit Error Rate (BER) due to poor signal-to-noise ratio (SNR). In this scenario, if a Wi-Fi device encounters interference from a Bluetooth transmission and subsequently slows its transmission rate, it will then spend more time than before transmitting a packet on a frequency available to Bluetooth, thus having the effect of increasing the likelihood of interference between the two. Data is not lost, but the data throughput rate may slow to an intolerable level.

The full impact of Wi-Fi interference depends on the utilization and proximity of Bluetooth devices. Interference can only occur when both Bluetooth and 802.11b devices transmit at the same time. However; some Bluetooth applications, such as printing from a laptop or synchronizing a PDA to a desktop, only utilize the radio for a very short period of time. In this case, the Bluetooth devices will generally not be active long enough to noticeably degrade the performance of an 802.11 network and vice versa. For example, a user may synchronize their PDA to their desktop when arriving at work in the morning. Other than that, their Bluetooth device may be inactive the rest of the day.

In addition to utilization, the proximity of the Bluetooth devices to 802.11 radio NICs and access points has a tremendous effect on the degree of interference. The transmit power of Bluetooth devices is generally much lower than 802.11 wireless LANs. Thus, an 802.11 station must be relatively close (within ten feet or so) of a transmitting Bluetooth device before significant interference can occur.

Figure 6(a): Wi-Fi Throughput with Bluetooth Interferer [5]

Figure 6(b): Bluetooth Throughput with Wi-Fi Interferer [5]

<u>Recommendations to Avoid Interference and Maximize Performance:</u>

The number of Bluetooth piconets in the same area BlueKey will be use in, should be less than 15. 15 or more piconets will start creating annoying interference and the user will receive 10% less data than expected.

In areas where 802.11 wireless LAN are setup, it is recommended that the user keep BlueKey off until it is needed to perform an operation and then turned off when the desire operation is completed.

Consider using 5 GHz NICs and access points for your 802.11 wireless LAN. You can completely avoid RF interference in this band. You'll also receive much higher throughput; however, the limited range could require additional access points and higher costs.

Locate if possible the 802.11 station more than ten feet away from where BlueKey may be use to avoid significant interference.

Project Hardware Technical Description

Written by Matthew Martinez

Block Diagram

Description

The hardware for the Personal Identification Device (PID) is organized into three sections: the Bluetooth module, the flash memory, and the power system. The Bluetooth module that will be utilized for this project is the Mitsumi WML-C10 Bluetooth module, which has an attached antenna. The WML-C10 is capable of transmitting up to 30 feet (10 meters) and possesses multiple programmable I/O pins which will be used to interface with the flash memory. The WML-C10 will function as the microcontroller of the PID as well as the means to transmit wirelessly using the Bluetooth protocol. The Atmel AT45DB642 is the flash memory component that will be utilized for the project. This flash memory component is capable of storing 64 MB of data. For easy data storage and access the memory is equipped with two 1 KB buffers and the ability to transmit and receive data either through serial communication or parallel communication. The memory will interface with the Bluetooth module through the parallel I/O pins. The external memory will have two main purposes. The first purpose is to store Bluetooth Mac address of the client computer with personal data that corresponds to the automation features. The second purpose of the external memory is to store personal data of any type, allowing BlueKey to act as a personal thumb drive which gives BlueKey more functionality and meaningful purpose to the consumer. The power system consists of the power source and the necessary voltage and current regulation circuits. The PID will be powered by two CR2016 coin batteries placed in series for a total of six volts. The Bluetooth module and flash memory have very specific power requirements which will be met through the use of two ST Microelectronics L200 voltage regulators. The ST Microelectronics L200 is a programmable voltage regulator that is also cable of programmable current limiting; programming occurs through the amount of resistance applied to certain pins. This voltage regulator also provides short circuit protection and thermal overload protection. The voltage regulator attached to the Bluetooth module will be program so that its maximum voltage output will be 3.3 V and its maximum current output will be 120 mA. Since the Bluetooth module has two different voltage requirements the output of its L200 voltage regulator will be split in parallel and one line will have an OPAMP voltage divider to meet the lower voltage requirement. The split of the output of the voltage regulator should reduce the maximum current output on each line to 60 mA. The voltage regulator attached to the flash memory will be programmed so that its maximum voltage output will be 2.85 V and its maximum current output will be 20 mA. An analog switch will be attached to the power source, permitting the user to turn the PID on and off; a green LED will be connected to the other end of the switch to act as a power indicator. Since the PID does not require any large circuitry, it should be a small portable device.

Preliminary Circuit Design

Software Design Concept

Written by Steve Wu

Computer Software Flowchart

Computer Software Case Studies
Written by Steve Wu

BlueKey Setup Module: Use Case

Use Case Section
Comments
Name
Automatic Browser Control
Scope
BlueKey Control Application
Level
User-goal
Primary Actor
General User
Steak-holders and Interest
General User – The robustness of the software affects how the user sets up the software and how he/her could setup the Bluetooth module to work in the autonomous reaction system.
Preconditions
User must have a computer with Bluetooth capabilities and the module must be ready to accept setup commands and data, the module must also be powered on.
Success Guarantee
The user will feed the application setup information to the application, where it will relay relevant information to the module, such as the MAC address it should be establishing a connection with.
Main Success Scenario
General user submits information for the computer workstation they want the software to control and the Bluetooth module that will trigger reactions, to the application where is it process and loads appropriate data to the module to through some communication medium.
Special Requirements
-The text must be visible from 1 meter from the monitor.
-The computer must be turned on
- The computer has some reliable connection to the module
Frequency of Occurrence
Infrequent/ Rare

BlueKey Browser Control System: Use Case

Use Case Section
Comments
Name
Automatic Browser Control
Scope
BlueKey Control Application

Level
User-goal
Primary Actor
General User
Steak-holders and Interest
General User – The robustness of the software affects how the user sets up the software and how he/her could setup an automated task (Browser Control)
Preconditions
User must have a computer with Bluetooth capabilities, a web browser that can be controlled by the application, and sufficient user access to setup a Bluetooth connection and setup a serial device link. The module itself is setup to connect only to a specific Bluetooth device, defined by a MAC address, and contains the data needed for the software to work properly already pre-loaded on it.
Success Guarantee
Once the device is detected the software will get the URL information from the module and control the browser by performing and action such as going to specific website URL, all occurring with no direct user interaction.
Main Success Scenario
General user turns on the module and the computer station loads up a specific webpage.
Special Requirements
-The text must be visible from 1 meter from the monitor.
-The computer must be turned on
- The computer has the proper MAC address that is being searched by the module.
- User has already setup the module with relevant data to make to make the software work.
Frequency of Occurrence
Frequent

Software Design Layout
Written by Steve Wu

Service

```
Sub LoadInstructionSet
    Open XML File
    Load Instructions into Map-like data structure

Sub CheckForDevice
    if serial connection is open
            check for module by requesting ID
            if ID is “ID_RSP_BK1\r\n”
                    Then return true
                    Else return false

Thread
    System.Process.exec(Instruction, parameters)
    Terminate Thread
```

```
Sub Main
    LoadInstructionSet
    Loop forever
    If CheckForDevice
            If there is a valid device (CheckForDevice==true)
                    Create a new Thread and launch instructions from Instruction Set
            Else
                    Continue loop
```

Configuration Tool

```
Sub Main
    Check for events
            if Save Button event is triggered
                    save values from controls to XML file
            if Load event is triggered
                    open XML file
                    load values into controls (i.e. TextBoxes, ComboBoxes, and
                        CheckBoxes)
```

Bluetooth Module Software Flowchart
Written By Steve Wu

Testing Plan Overview

Written by Pablo Rosado

The testing of BlueKey will consist of two phases: Hardware and Software tests. Hardware testing will mainly be base on the power system and the individual hardware components to make sure they are not damage. Already, we have done computer simulations of the power system which is the first part of our hardware testing and plan on ordering extra parts in case individual components become damage during the testing phase. Once we receive the parts of the power system and determine that all the parts are the correct parts we order and are not damage, we will assemble the power system and simulate it on a breadboard and measure the voltage and current outputs. Note that during this testing phase we will not be connecting the Bluetooth module or the external memory since they are the most expensive parts of our project and would be difficult to replace due to time and cost if they get damage. If the power system test proves that it will not harm the Bluetooth module or the external memory and provide the proper power requirements to both hardware devices, then we can connect the Bluetooth module and the external memory and test if both hardware components can run using our power systems. Currently, we already have the Bluetooth module in our possession and already tested it to see if it can connect to a computer wirelessly. The remaining project which we would need to test is the software which will control the Bluetooth module, the external memory, and the computer communicating to the Bluetooth module. The software testing will be broken into three parts. The first part will be testing to see if the Bluetooth module will be capable to communicate with the computer and perform simple commands. The second phase of the testing is to test if the module will be capable of communicating to the external memory and perform simple tasks such as reading, writing, and deleting data form the external memory. If both these test are successful, then we can integrate both tests and test if the Bluetooth module is capable of reading data from the external memory and sending that data to the computer successfully and vise versa (the computer sending data to the Bluetooth module which it will interpret and perform the correct operation to the external memory.). After this we will be able to do more complex test to test every aspect of the BlueKey and make sure it will be able to do everything we specified in our requirements such as reading alternate Bluetooth Mac address from the external memory and connect to them and perform the correct task to that specific Bluetooth enable-device.

Technical and Schedule Risks and Solutions

Written by Matthew Martinez

Technical Risks and Solutions

Expanding the module's memory may not be easily implemented; thus the system will then have different requirements for the embedded software and a change in the power system.

  The embedded software will need to change and the power system will need to be accordingly adjusted, this can produce significant delays.

Finding a suitable simulation model for some of the components within the power system has proven difficult.

  If a suitable simulation model for these components cannot be found, it will be necessary to try different simulation software.

Interfacing with the Windows API has proven more difficult than originally anticipated.

  Research is currently being conducted to find solutions to the difficulties.

Schedule Risks and Solutions

Expanding the module's memory may not be easily implemented and the resulting need to make adjustments in design can result in significant delays.

  A schedule has been allocated that should account for such delays.

Finding solutions for interfacing with the Windows API could produce significant delays since it is necessary to complete Phase One testing before the module is shipped for reprogramming.

Construction components may arrive late from vendors.

  Will order components well in advance to accommodate the occurrence of such delays.

Possibility that access to equipment and resources necessary for production may be unavailable when needed.

  Will gain access to such equipment and resources before they are actually needed.

Schedule Status

Written by: Steve Wu

Fall 2005

Spring 2006

Labor Status

Written by: Matthew Martinez

Bill of Material

Written By Steve Wu

Line Item

Part / Model Number

Part Description

Manufacturer

Supplier / Vendor

Quantity

Unit of Measure (ea, bag, ft, etc)

Unit Price

Extended Price

1

P138-ND

BATTERY LITHIUM COIN 3 VOLT 20MM

Panasonic – BSG

Digi-Key

2

$0.30
$0.60

2
497-3068-5-ND
IC OPAMP SNGL MOS-INPUT 8-SOIC
STMicroelectronics
Digi-Key
2

$2.01
$4.02

3
401-1000-1-ND
SWITCH SLIDE DPDT 12V 100MA GW
ITT Industries
Digi-Key
2

$1.35
$2.70

4
AT45DB642-TI-ND
IC DATAFLASH 64M 2.7V 40TSOP
Atmel
Digi-Key
2

$18.10
$36.20

5
1026K-ND
Coin battery holder:
Keystone Electronics
Digi-Key
2

$1.03
$2.06

6
CCM05-5501CT-ND
Memory card holder
ITT Industries
Digi-Key
2

$1.43
$2.86

TOTAL:

$48.44